PUBLICATION OF PATENT APPLICATION

(11) Publication number: 2001-203749
(43) Date of publication of application: July 27, 2001
(51) Int.Cl. H04L 12/56
(21) Application number: 2000-11533 (P2000-11533)
(22) Date of filing: January 20, 2000
(71) Applicant: Mitsubishi Electric Co.
(72) Inventor: Yuji Nimura

(54) Highly-efficient data transmission apparatus, and highly-efficient data transmission system

(57) **Abstract:**

[PROBLEM TO BE SOLVED]: To improve a transmission efficiency by reducing an overhead of divided transmissions by forcibly transmitting a transmission unit in a short time, and collecting short data pieces.

[SOLUTION]: A configuration for transmitting data that are input from an inputting means by making the data redundant, and encoding, the configuration including a buffer for storing the data in units of a predetermined length, a data storage/forced transmission directing means for directing to forcibly transmit (S15) by dividing and encoding the data when a predetermined time has elapsed (S14) from a moment when the data are stored in the buffer that is vacant, and a data encoding/dividing means for making data in the predetermined length stored in the buffer redundant and for encoding the data.

[Translation of Drawing in ABSTRACT]

S10 START

S11 DATA RECEIVED FROM DATA INPUTTING MEANS

S12 DATA STORED IN BUFFER

S13 HOLD DATA RECEIVING TIME

S14 HAS TIME T ELAPSED SINCE DATA RECEIVING TIME IS STORED?

S15 PROVIDE DATA STORED IN BUFFER TO DATA ENCODING/DIVIDING MEANS

S16 BUFFER EMPTIED

S17 ARE NEW DATA PRESENT IN DATA INPUTTING MEANS?

S18 DATA RECEIVED FROM DATA INPUTTING MEANS, AND ADDED TO IN BUFFER

What is claimed is:

[Claim 1] A high-efficiency data transmission apparatus, wherein transmitting data that are input from an inputting means are made redundant, encoded, and transmitted in packets, the transmission apparatus comprising:

a buffer for storing the data in units of a predetermined length;

a data storage/forced transmission directing means for directing forced transmission of the data that are divided and encoded when a predetermined time has elapsed from a moment when the data are stored in the buffer that is vacant; and

a data encoding/dividing means for making the data redundant and for encoding the data, the data being stored in the buffer in the predetermined length.

[Claim 2] The high-efficiency data transmission apparatus as claimed in Claim 1, wherein the data encoding/dividing means determines the number of divisions of the data length, and a degree of the redundancy based on the predetermined length of the transmitting data and a transmission probability that is acquired in an allowable

transmission delay time when the predetermined time has elapsed.

[Claim 3]

The high-efficiency data transmission apparatus as claimed in Claim 1, wherein the data storage/forced transmission directing means dynamically changes the predetermined time based on a transmission probability that is acquired in an allowable transmission delay time.

[Claim 4] The high-efficiency data transmission apparatus as claimed in one of Claims 2, and 3, further comprising a packet receiving means for acquiring delay time information, and calculating the transmission probability.

[Claim 5] The high-efficiency data transmission apparatus as claimed in Claim 1, wherein the data that are input include an identifier to indicate whether division/combination is possible; and, if division/combination is possible, the data are stored at the predetermined length of the buffer, are made redundant, and are encoded.

[Claim 6] The high-efficiency data transmission apparatus as claimed in Claim 1, wherein the data that are input include an identifier to identify a stream, the data are stored in the buffer by the stream identifier,

when the length of the data reaches the predetermined length, the data are made redundant and encoded for transmission.

[Claim 7] A high-efficiency data transmission system, comprising:

a data transmission apparatus; and

a data receiving apparatus;

the data transmission apparatus, comprising:

a buffer for storing data provided by an inputting means in units of a predetermined length;

a data storage/forced transmission directing means for directing forced transmission of the data that are divided and encoded when a predetermined time has elapsed from a moment when the data are stored in the buffer that is vacant; and

a data encoding/dividing means for making the data redundant and for encoding the data, the data being stored in the buffer in the predetermined length; and

the data receiving apparatus, comprising:

a data decoding means for receiving the data from the transmission apparatus via a communication line, and immediately starting decoding when data of a predetermined degree of redundancy are received.

[Detailed Description of the Invention]

[0001]

[Field of the Invention] The present invention generally relates to a technology for enhancing reliability of communications in a communication line for packet data communications, and especially relates to a technology for enhancing quality and efficiency of the communications wherein losses are reduced, fluctuation of a required communication time is reduced, and the communication time is reduced in a network where packet transfer is not highly reliable such as the Internet wherein the required communication time is highly variable.

[0002]

[Description of the Prior Art] In a communication line wherein data communications are

carried out by packets, in order to enhance the reliability of the communications, a technique generally used is as follows: a transmitter transmits redundant data that are divided into two or more packets, and a receiver receives only a part of the packets, yet recovers the original data, that is, all packets are not necessarily required to be received.

[0003] For example, JPA H8-228190 discloses "a forward error correction system, and a method thereof", which is referenced as the first conventional example, and described using Fig. 17.

[0004] Fig. 20 shows the configuration of an embodiment of the first conventional example. In the drawing, a transmission apparatus 101 and a receiving apparatus 201 exchange data through data terminal equipment (DTE) 104, data communication equipment (DCE) 105, data terminal equipment (DTE) 204, and data communication equipment (DCE) 205. The transmission apparatus 101 and the receiving apparatus 201 are realized by common electronic components. The electronic components include microprocessors 110/210, digital signal processors 111/211 connected to the microprocessors, and controllers 112/212 for synchronizing a timing signal between the components. The controllers 112/212 further monitor the microprocessors 110/210 and the digital signal processors 111/211 such that packets are duly transmitted and received.

[0005] The microprocessors 110/210 include central processing units (CPU) 113/213, EEPROM (electrically Erasable Programmable ROM) 114/214, and RAM (Random Access Memory) 115/215. The CPU 113 executes a program stored in EEPROM 114 so that data are packetized (packet assembly), framed, and encoded; further, in order to form M parity packets, N packets are encoded by combining a few packets selected out of N original packets, and a data stream

consisting of N+M packets is transmitted to the receiver 201. The CPU 213 executes a program stored in EEPROM 214 so that the data packets are disassembled, decoded, and the N original packets are recovered so long that at least N packets out of the N+M packets in the data stream are properly received with no damage. Here, the values of N and M do not have to be fixed, but may be dynamically adjusted according to the number of errors detected by the receiving apparatus 201.

[0006] According to the first conventional example, M parity packets are generated to N packets, and N+M packets constituting the data stream are transmitted from the transmission apparatus 101 to the receiving apparatus 201; for this reason, the data stream cannot be transmitted from the transmission apparatus 101 to the receiving apparatus 201 until all the N packets are prepared on the side of the transmission apparatus 101. That is, a packet that has earlier arrived at the DTE 104 cannot be transmitted until all the N packets arrive, which generates a transmission delay. Further, when generating the M parity packets from the N packets, it is presumed that the size of the packets are the same; accordingly, if the sizes of the packets are different, the sizes have to be aligned to the longest (plus a margin if required) of the N packets. This generates a useless traffic due to greater than necessary packet number N.

[0007] The second conventional example, JPA H9-64913 "A data assurance method of packet communications", attempts to solve this problem. This is described with reference to Fig. 18. Fig. 21 shows the system configuration of the second conventional example. At (a), a block diagram of a data communication system according to the second conventional example is given; and at (b) a memory

structure of the example is illustrated. As shown in Fig. 21(a), an information processing apparatus 300 includes a CPU 301 for operation processes, a memory 302, and a communication network interface (I/F) 302 for connecting to a communication network 304 so that communications with other information processing apparatuses 300 are enabled.

[0008] Fig. 21(b) shows the structure of the memory 302 of the information processing apparatus 300. As illustrated, three each of fragments 311 and 312 are provided to a transmission packet 310 and receiving packet 320; further, fragments for recovering 312 and 322 are provided. Here, the size of the fragments 311, 321, and the fragment for recovering 321 and 322 is equal to or less than the maximum data length that can be transmitted/received by the communications network 304. Three fragments 311 are generated by dividing contents of the transmission packet 310 into three parts. The fragment for recovering 312 is generated by a predetermined logical operation of the 3 fragments 311 such that the contents of the transmission packet 310 can be recovered even if one of the 3 fragments 311 and the fragment for recovering 312 is lost.

[0009] As described above, the information processing 300 generates the fragments 311 and the fragment for recovering 312 from the transmission packet 310 and transmits them to other information processing apparatuses 300. The information processing apparatuses 300 can recover the original contents of the transmission packet 310 as the receiving packet 320 if all the fragment 321 are received, or two out of three fragments 321 and a fragment for recovering 322 are received. Here, the case wherein the contents of the transmission packet 310 is divvied into three parts is described;

nevertheless, the contents can generally be divided into N parts, wherein the value of N may be dynamically adjusted according to a loss rate of the fragments in the communications network 304.

[0010] According to the second conventional example, unlike the first conventional example, the data are made redundant based on one packet; for this reason, the problem of the first conventional example, i.e., the data transmission delay is solved. However, according to the second conventional example, even if the size of the transmission packet 310 before being divided into the fragments 311 is equal to or less than the maximum that can be transmitted/received by the communications network 304, the packet is divided into fragments for transmission. According to the second conventional example, data are divided into a predetermined number of parts. For this reason, if this technique is used by a high-speed communication network that is widely used in recent years, since short data are always divided into parts, a great overhead is generated. Thus, the number of fragments influences the traffic, that is, an increased number of fragments to be transmitted require useless time, degrading the transmission efficiency. This problem can easily arise if this technique is applied to IP packets generated based on TCP/IP protocol used in the Internet for an Ethernet LAN capable of 100 Mbps or 1 Gbps structured by switching hubs.

[0011]

[Problem(s) to be solved by the Invention] The problems of the conventional techniques to be solved, in summary, are that if the degree of redundancy is raised to secure data contents, it takes time to receive a required number of packets (the first conventional example), and that if data are divided, the transmission efficiency is degraded (the second conventional example).

[0012] The present invention is made in order to solve the problems described above, wherein a transmission unit is forcibly transmitted in a short time lapse, and short data are collected and transmitted; in this way, data transmission delay and generation of useless traffic are reduced; and an overhead time of divided transmission is small, enhancing the communication quality and transmission efficiency.

[0013]

[Means for solving the Problem] The high-efficiency data transmission apparatus according to the present invention makes data from an inputting means redundant, encodes the data, and transmits the data in packets, and the high-efficiency data transmission apparatus comprises a buffer for storing the data in units of a predetermined length, a data storage/forced transmission directing means for directing forced transmission of the data that are divided and encoded when a predetermined time has elapsed from a moment when the data are stored in the buffer that is vacant, and a data encoding/dividing means for making the data redundant and for encoding the data, the data being stored in the buffer in the predetermined length.

[0014] Further, the data encoding/dividing means determines the number of divisions of the data length, and a degree of the redundancy based on the predetermined length of the transmitting data and a transmission probability that is acquired in an allowable transmission delay time when the predetermined time has elapsed.

[0015] Further, the data storage/forced transmission directing means dynamically changes the predetermined time based on a transmission probability that is acquired in an allowable transmission delay time.

[0016] Further, a packet receiving means is provided such that delay time information is acquired, and the transmission probability can be calculated.

[0017] Further, an identifier that indicates whether dividing/combining is possible is attached to the data that are input; and if dividing/combining is possible, the data are stored in the predetermined length units of the buffer, made redundant, and encoded.

[0018] Further, an identifier that identifies a stream (stream identifier) is attached to the data that are input; the buffer stores the data according to the stream identifier; if data length reaches the predetermined length, the data are made redundant and encoded for transmission.

[0019] The high-efficiency data transmission system according to the present invention comprises:

a data transmission apparatus; and

a data receiving apparatus;

the data transmission apparatus, comprising:

a buffer for storing data provided by an inputting means in units of a predetermined length;

a data storage/forced transmission directing means for directing forced transmission of the data that are divided and encoded when a predetermined time has elapsed from a moment when the data are stored in the buffer that is vacant; and

a data encoding/dividing means for making the data redundant and for encoding the data, the data being stored in the buffer in the predetermined length; and

the data receiving apparatus, comprising:

a data decoding means for receiving the data from the transmission apparatus via a communication line, and immediately starting decoding

when data of a predetermined degree of redundancy are received.

[0020]

[Embodiment of the Invention]

Embodiment 1

A system wherein fragmentary packetization of data is reduced, and a standby time before starting transmission is reduced is described. Fig. 1 shows the system configuration according to Embodiment 1 of the present invention. In Fig. 1, a data transmitting means 1, and a data receiving means 2 are connected by a communication line 3. The data transmitting means 1 includes a data inputting means 4, a data storing/forced transmission directing means 5, which is a novel and important element, a buffer 6, a data encoding/dividing means 7, and a packet transmitting means 8. The data receiving means 2 includes a packet receiving means 9, a data decoding means 10, and a data outputting means 11.

[0021] Next, operations of the system according to the present embodiment are described with reference to Fig. 2. In Fig. 2 for describing the operations, in the data transmitting means 1, one or more sets of data 20 received by the data inputting means 4 are stored in the buffer 6 at directions of the data storing/forced transmission directing means 5. The data storing/forced transmission directing means 5 provides the data encoding/dividing means 7 with the data 20 when a time T has elapsed since when the oldest of the sets of the data 20 was stored. The data encoding/dividing means 7 divides the data 20 that are provided by the data storing/forced transmission directing means 5 into N packets 21 (N=2 in Fig. 2), and generates by encoding M packets 21 (M=1 in Fig. 2) for holding parity information such that the original data 20 may be recovered if only N packets out of N+M packets 21

are acquired. Then, the data encoding/dividing means 7 transmits the N+M packets 21 using the packet transmitting means 8 to the communication line 3.

[0022] The N+M packets 21 transmitted through the communication line 3 are received by the packet receiving means 8 of the data receiving means 2, and provided to the data decoding means 10. The N+M packets 21 that are transmitted by the packet transmitting means 8 are received not necessarily in the same order as transmitted; times required for transmission of the packets are not constant, but vary; a part of the packets may be lost and therefore not provided to the data decoding means 10. When N packets 21 are provided by the data receiving means 2, the data decoding means 10 carries out a data decoding process, and recovered data 21 are output through the data outputting means 11, given that the original data 20 can be recovered from any N packets 21 out of the N+M packets 21.

[0023] Fig. 3 is a flowchart showing operations of the data storing/forced transmission directing means 5 according to the present embodiment. At the first processing step S10 carried out by the data storing/forced transmission directing means 5, the data 20 are received from the data inputting means 4 (S11), the received data 20 are stored in the buffer 6 (S12), and the time of receiving is temporarily stored (S13). If a time T has elapsed (Yes at S14), the data 20 stored in the buffer 6 are provided to the data encoding/dividing means 7 (S15), and the buffer 6 is cleared (S16). Then, the process returns to S11 for receiving new data from the data inputting means 4. If new data 20 are received (Yes at S17) while the time T has not elapsed (No at S14), the data are additionally stored in the buffer 6 (S18) until a data volume reaches a certain level.

[0024] Operations of the data

encoding/dividing means 7 and the data decoding means 10 according to the present embodiment are described with reference to Figs. 4, and 5. The data encoding/dividing means 7 may use any of various generally available methods of making data redundant. For example, where M=1, a parity method using an XOR logic. In the following, the case wherein a parity method, wherein N=2, and M=1, using the XOR logic is employed is described. As a unit of making data redundant, and dividing the data, various units can be used such as a bit, a byte, and 4 bytes. The present invention is independent of such units. Accordingly, for easiness of descriptions, the data are handled in the units of byte.

[0025] Fig. 4 is for describing the operations of the data encoding/dividing means 7 when the parity method using the XOR logic wherein N=2, and M=1 is employed. The data encoding/dividing means 7 divides the data 20, which are directed for transmission by the data storing/forced transmission directing means 5, into two parts (N=2), namely, partial data A 22, and partial data B 22, having the same size. Here, if the size of the data 20 is not an integer multiple of N, bytes having 0 value are beforehand added so that the data 20 can be divided by N with no remainder. Then, as for the N (=2) sets of the partial data 22, XOR of bits in the same position is acquired to generate parity data P 23 (Fig. 4(a)), having the same size as the partial data. Then, an ID number, total data volume, and values showing a starting position of each set of the partial data ("-1" in the case of the parity data) are added to the N (=2) sets of the partial data 22 (A and B), and the M (=1) set of the parity data P 23 to generate the packet 21 (Fig. 4(b)).

[0026] Operations of the data storing/forced transmission directing means 5

corresponding to the operations of the data encoding/dividing means 7 are described with reference to Fig. 5. A packet 21 having the same ID number out of the packets 21 provided by the packet receiving means 9 is a target of dada decoding process. If both packets 21 having the same ID number hold the partial data 22, the data 20 are recovered by simply connecting the partial data A 22, and B 22 (Fig. 5 (a)). If one of the two packets 21 having the same ID number holds the partial data 22, and the other holds the parity data 23, XOR of the partial data A 22 and the parity data P 23 at the same positions is obtained, and the partial data B 22 that are not received are reproduced. Then, the received partial data A 22 and the reproduced partial data 22 are connected to recover the data 20 (Fig. 5(b)).

[0027] The system according to the present embodiment operates as described above. Accordingly, if, for example, a probability of a packet transmitted by the transmitting means 8 is received by the receiving means 9 through the communication line 3 within a time Td is 0.9, the probability when the data are not made redundant and simply transmitted by two packets is 0.81, and the probability when the data are made redundant with N=2, and M=1 is improved to 0.972. In addition, processes of the data transmitting means 1 and the data receiving means 2 are not greatly complicated compared with the processes of the conventional examples 1, and 2, except that the data are temporarily held by the data storing/forced transmission directing means 5. Therefore, the processing time is not much different from the conventional techniques, and is sufficiently short with reference to a packet communication delay time of the communication line 3. Further, the data staying duration in the buffer 6 is T at the greatest.

For this reason, a time from the data being input to the data inputting means 4 to the data being obtained from the data outputting means 11 approximately is a sum of the communication delay time of the communication line 3, and the time T directed by the data storing/forced transmission directing means 5 at the greatest. The data transmission delay can be reduced by assuring, for example, that the data that are input to the data inputting means 4 are obtained from the data receiving means 11 within Td+T at the probability of 0.972.

[0028] Further, according to the present embodiment, the data acquired by the data inputting means 4 are stored in the buffer 6 by the direction of the data storing/forced transmission directing means 5, then provided to the data encoding/dividing means 7 to be encoded and packetized. Accordingly, the number of packets is less than the case wherein data are individually encoded and packetized without being stored in the buffer 6, that is, generating useless traffic is reduced.

[0029] Although the embodiment described above employs the parity method using the XOR logic, wherein N=2, and M=1 are used for redundancy used by the data encoding/divining means 7 and the data decoding means 10, the parity method using the XOR logic can be used with N>2, providing the same effect. Further, in the case where M>1, as a method wherein N and M are generalized, a Reed-Solomon method, and other methods can be used to obtain the same effect. Further, in the case that N=1, and M=1, redundancy can be obtained by copying, and the same effect can be obtained.

[0030] Although, according to the embodiment described above, the data 20 are divided into N sets of partial data 22, and then M sets of parity data 23 are generated; nevertheless, the data

20 can be divided into N+M packets after a redundancy process of making the data 20 to be dividable into N+M packets. In general, other methods of generating N+M packets 21, wherein if only N packets are obtained, the original data are recovered, may be used to obtain the same effect.

[0031] Fig. 6 is a flowchart showing operations of another data storing/forced transmission directing means 5b according to the embodiment. The data storing/forced transmission directing means 5b forcibly transmits based on the length of the buffer 6, in addition to the time T, as described below with reference to Fig. 6. The process step loop from S20 to S16 in Fig. 3 of the data storing/forced transmission directing means 5 applies to the data storing/forced transmission directing means 5b. Then, even if the time T has not elapsed (No at S14), if the size of the data 20 stored in the buffer 6 is greater than a predetermined value L (Yes at S21), the data 20 stored in the buffer 6 are provided to the data encoding/dividing means 7 (S15), the buffer 6 is cleared (S16), and the process returns to S11 where new data 20 are received from the data inputting means 4. Otherwise, if the time T has not elapses (No at S14), and if the size of the data 20 stored in the buffer 6 is less than L (No at S21), when new data 20 are received from the data inputting means 4 (Yes at S17), the new data 20 are additionally stored in the buffer 6 (S18).

[0032] That is, the size of the buffer 6 can be defined by a sum of the predetermined value L and the greatest size of data 20 that may be provided by the data inputting means in one shot. Accordingly, the greatest size of the data 20 that may be provided to the data encoding/dividing means 7 is limited, which enhances the efficiency of processing. Further,

if frequency of data inputting to the data inputting means 4 is great, the data transmission process is carried out before the time T has elapsed.

[0033] If the unit of data collection into one packet is short, the overhead is increased and the dividing loss is generated as in the second conventional example; if it is long, it takes time to finish the transmission. Further, data lengths are not always appropriate. According to the embodiment, the buffer length is determined to an optimal packet length, and data are collected in that unit. Fig. 7 is a flowchart showing operations of further another data storing/forced transmission directing means according to Embodiment 1. In the following, the operations of the means other than the data storing/forced transmission directing means 5 are the same as described with reference to Figs. 1 through 5 described in Embodiment 1, and therefore, descriptions are not repeated.

[0034] With reference to the flowchart in Fig. 7, the data 20 acquired by the data inputting means 4 are stored in the buffer, the length of which buffer is predetermined, and when the data are stored in the buffer, transmission is started. Specifically, a process of the data storing/forced transmission directing means 5c starts with a process S30, the data 20 are acquired from the data inputting means 4. If the buffer 6 is vacant (Yes at S31), a time at which the data 20 are received is held (S13). Then, then data 20 are received so long as vacancy of the buffer 6 is available (S31). When the volume of the data 20 stored in the buffer 6 exceeds the predetermined value L (Yes at S21), or if the time T has elapsed since the time that is held even if the volume of the data 20 stored in the buffer 6 does not exceed L (No at S21), the data 20 stored in the buffer 6 are provided to the data encoding/dividing

means 7 (S15), and the buffer 6 is cleared (S16).

[0035] If there is a part of the data 20 that are received but not yet stored in the buffer 6 (Yes at S33), the part is regarded as the data 20 newly received by the data inputting means 4, and the process is repeated (S31). If all the received data 20 are stored in the buffer 6 (No at S33), when new data 20 are provided by the data inputting means 4 (Yes at S17), the data 20 are newly received from the data inputting means 4 (S11), and the process is continued. Then, if all the received data are stored in the buffer 6 (No at S33), and there are no new data 20 from the data inputting means 4 (No at S17), the process returns to S14 wherein whether the time T has elapsed is determined.

[0036] As described above, the size of the buffer 6 can be defined by the predetermined value L, and the size of the data to be provided to the data encoding/dividing means 7 can be limited to the above size at the maximum. Accordingly, it is possible to appropriately limit the size of a work area required by the processes carried out by the data storing/forced transmission directing means 5 and the data encoding/dividing means 7. In this way, the process efficiency is improved, and where the frequency of data inputting to the data inputting means 4 is high, the data can be transmitted before the fixed time T has elapsed.

[0037]

Embodiment 2

It is desired that the length of the data transmission unit is adjustable depending on congestion and quality of the communication line; that is, the data length of the transmission unit is decreased if many errors arise, and the data length is increased to avoid the dividing loss if errors are few. In this case also, a time limit is provided in

order to shorten the completion time. Fig. 8 is a flowchart showing operations according to Embodiment 2. That is, a method of determining the values of N, and M is shown, when the data 20 provided by a data storing/forced transmission directing means 5d are encoded and divided into N+M packets 21 by the data encoding/dividing means 7. The system configuration and operations according to Embodiment 2 are the same as those of embodiment 1 shown in Figs. 1 through 5, except for the method of determining the values of N, and M; accordingly descriptions thereof are not repeated.

[0038] The process is started at S40. At the following S41,

based on an allowable greatest delay time Th, and an achievement probability Ph thereof,

in consideration of the longest time T that the data 20 may stay in the buffer 5 as directed by the data storing/forced transmission directing means 5,

from the distribution of packet arrival ratio corresponding to communication delay time in the communication line 3,

a packet arrival ratio Pd of receiving the packet within the communication delay time Th-T is acquired.

Next, from a length Ld of the data 20 provided by the data storing/forced transmission directing means 5, and a length Lp that is the greatest length of data that can be contained in the packet 21 transmitted by the packet transmitting means 8 to the communication line 3, a minimum number Nmin that is equal to or greater than Ld/Lp is acquired (S42).

Next, N and M are obtained (S43), wherein N+M takes a value that is as small as possible, while N/(N+M) is as great as possible, and sufficing

N>=Nmin, M>=1, Ph<= $\sum_{i=0}^{M}({}_{N+M}C_i \times (1-Pd)^{i} \times Pd^{n+m-i})$. Here, $\sum_{i=0}^{M}({}_{N+M}C_i \times (1-Pd)^{i} \times Pd^{n+m-i})$ represents a probability of N or more packets 21 out of N+M packets 21 transmitted are received within the time Th-T. Further, where the size of the packet, rather than the number of the packets, has a greeter influence to the traffic of the communication line 3, the values of N and M are determined by evaluating N/(N+M) more importantly than N+M. Conversely, where the number of the packets, rather than the size of the packet, has the greater influence to the traffic of the communication line 3, N+M is evaluated more importantly than N/(N+M). Then, based on the values of N and M obtained as described above, the data encoding/dividing means 7 encodes and divides the data 20 that are provided by the data storing/forced transmission directing means 5, to generate the packet 21.

[0039] As the system according to the present embodiment operates as described above, the quality and efficiency of data communication are improved sufficing the allowable maximum delay time Th, and its achievement probability Ph, based on the distribution of the packet arrival rates corresponding to the communication delay time in the communication line 3, the size of the data 20 provided by the data storing/forced transmission directing means 5, and the greatest size of data that can be contained in the packet to be transmitted.

[0040] The distribution of the packet arrival rates corresponding to the communication delay time in the communication line 3, which is used in the embodiment, may be obtained by measuring communication characteristics of the communication line 3 apart from the process of the embodiment.

Alternatively, the distribution of the packet arrival rates corresponding to the communication delay time of the communication line 3 can be served by measuring the communication delay time of the packets 21 by the packet receiving means 9 when the packets 21 are transmitted by the packet transmitting means 8 of the data transmitting means 1 to the packet receiving means 9 of the data receiving means 2, and by providing statistical information acquired from measurement results from the data receiving means 2 to the data transmitting means 1.

[0041] Fig. 9 is a flowchart showing operations of adjusting the maximum time T, during which T the data 20 are stored according to another data storing/forced transmission directing means 5e of Embodiment 2. The data storing/forced transmission directing means 5e carries out the following process based on the allowable maximum delay time Th, and its achieving probability Ph. In the following, it is assumed that the values of N, and M are predetermined; here, N and M are for encoding and dividing the data 20 provided by the data storing/forced transmission directing means 5 in the data encoding/dividing means 7.

[0042] The process starts at S50, and Pd that satisfies $Ph=\sum_{i=0}^{M}({}_{N+M}C_i \times (1-Pd)^i \times Pd^{n+m-i})$ is acquired (S51). Here, $\sum_{i=0}^{M}({}_{N+M}C_i \times (1-Pd)^i \times Pd^{n+m-i})$ represents a probability of N or more packets 21 out of N+M packets 21 transmitted are received, wherein Pd is an arrival rate of one packet. Next, from the distribution D(τ) of packet arrival rates corresponding to the communication delay time in the communication line 3, t that satisfies D(Th-t)>=Pd is acquired. The acquired t is made into the maximum time T during which the data 20 are held as directed

by the data storing/forced transmission directing means 5e (S52).

[0043] As described, the efficiency of encoding the data is enhanced by satisfying the allowable maximum delay time Th and its achieving probability Ph based on the distribution of the packet arrival rates corresponding to the communication delay time of the communication line 3, and by storing the data as long as possible by the data storing/forced transmission directing means 5e.

[0044] Although the descriptions above are made in the premise that N and M used by the data encoding/dividing means 7 for encoding and dividing the data 20 provided by the data storing/forced transmission directing means 5 are given; nevertheless, they can be dynamically adjusted according to the time T such that the effect of the present invention is further improved.

[0045] Further, as the distribution of the packet arrival rates corresponding to the communication delay in the communication line 3 used in the above, instead of the arrival information from the counterpart, receiving information of a receiving circuit on the transmission side can be used to obtain almost the same result. Fig. 10 shows such a configuration. The configuration includes

a first data transmitting means A 1b,

a first data receiving means A 2b that is paired with the first data transmitting means A 1b,

a second data receiving means B 2c that exchanges data with the first data transmitting means A 1b through the communication line 3, and

a second data transmitting means B 1c that is paired with the second data receiving means B 2c for exchanging data with the first data receiving means A 2b through the communication line 3. Here, data inputting means A 4/B 4c, data storing/forced

transmission directing means A 5/B 5c, buffers A 6/B 6c, data encoding/dividing means A 7/B 7c, and packet transmitting means A 8/B 8c of the data receiving means A 2b/B 2c; and packet receiving means A 9/B 9c, data decoding means A 10/B 10c, and data outputting means A 11/B 11c of the data receiving means A 2b/B 2c are the same as shown in Fig. 1. Accordingly, detailed descriptions of these items are not repeated.

[0046] According to the present system, the packet receiving means A 9 of the data receiving means A 2b measures a communication delay time of each of the packets 21 that are transmitted by the packet transmitting means B 8c of the data transmitting means B 1c. Then, statistical information obtained from results of the measurement is provided to the data encoding/dividing means A 7 of the data transmitting means a 1b. The data encoding/dividing means A 7 of the data transmitting means a 1b regards the statistical information provided by the packet receiving means A 9 as the distribution of packet arrival rates corresponding to the communication delay time on the transmission side of the communication line 3, and adjusts the values of N and M that are used when encoding and dividing the data 20 provided by the data storing/forced transmission directing means A 5 into N+M packets 21.

[0047] In this way, the data transmitting means A 1 can obtain the distribution of packet arrival rates corresponding to the communication delay time of the communication line 3 by its receiving side, and obtain appropriate values of N and M according to the communication quality of the communication line 3 without needing statistical information about the communication delay time of each packet from the data receiving means B 2 of the communication counterpart.

[0048] In addition, the concept of the

present embodiment is applicable to the case wherein the data transmitting means 1 communicates with two or more data receiving means 2. That is, Fig. 11 shows the case wherein a data transmitting means 1d communicates with plural data receiving means; namely, the data transmitting means 1d exchanges data with a data receiving means A 2e via a communication line A 3a, and simultaneously exchange data with a data receiving means B 2f via a communication line 3b. Here, the data inputting means 4, a data storing/forced transmission directing means 5f, the buffer 6, the data encoding/dividing means 7, and the packet transmitting means 8 of the data transmitting means 1d; and the packet receiving means A 9/B 9, data decoding means A 10/B 10, and data outputting means A 11/B 11 of the data receiving means A 2e/B 2f are the same as shown in Fig. 1.

[0049] According to the configuration described above, the data encoding/dividing means 7b, when exchanging data with the data receiving means A 2e via the communication line A 3, adjusts the values of N and M to be used when encoding and dividing the data 20 directed by the data storing/forced transmission directing means 5f into N+M packets 21 using the maximum data size Lpa that can be contained in the packet 21 transmitted by the packet transmitting means 8 to the communication line A 3, and the packet arrival rate Pda of packets being received within the communication delay time Th-T that is obtained from the distribution of the packet arrival rates corresponding to the communication delay time of the communication line A 3; and the data encoding/dividing means 7b, when exchanging data with the data receiving means B 2 via the communication line B 3, adjusts the values of N and M to be used when encoding and dividing the data 20 directed by the data storing/forced transmission

directing means 5f into N+M packets 21 using the maximum data size Lpb that can be contained in the packet 21 transmitted by the packet transmitting means 8 to the communication line B 3, and the packet arrival rate Pdb of packets being received within the communication delay time Th-T that is obtained from the distribution of the packet arrival rates corresponding to the communication delay time of the communication line B 3.

[0050] According to the present system, the values of N and M are appropriately selected to each of the communication lines A 3a/B 3b according to communication characteristics; therefore the quality and efficiency of data communications are improved.

[0051] Embodiment 3

The case wherein short data are to wait until reaching a predetermined buffer length, and a division loss is avoided. Fig. 12 is a flowchart showing operations of the data storing/forced transmission directing means 5 according to the present embodiment. The system configuration of the present embodiment is the same as shown in Fig. 1.

[0052] According to the present embodiment, the data inputting means 4 receives the data 20 as packets, and stores the data 20 in the buffer 6 as they are. A data storing/forced transmission directing means 5g carries out processes starting at S60, receives the data 20 (S11), and holds the data receiving time (S13) if the buffer was empty (Yes at S31). Then, kinds of the received data 20 are determined (S61). If the received data 20 are packets that cannot be divided/combined (No at S61), the remaining capacity of the buffer 6 is determined, and if all the data 20 can be stored in the buffer 6 (Yes at S62), the received data 20 are stored in the buffer 6 (S63). Otherwise, if the received data 20 are packets that can be divided/combined (Yes at S61),

packets stored in the buffer 6 are examined, and if packets that can be combined are present in the buffer 6 (Yes at S64), the received data 20 are stored in the buffer 6, being combined with the existing packets up to the capacity of the buffer (S65); if not (No at S61), the data 20 are stored in the buffer 6 as individual packets up to the capacity of the buffer 6 (S32).

[0053] Here, a packet that can be divided/combined is a packet that holds data in a stream form (described below) that does not require to conserve data boundary, and holds contiguous data belonging to the same data stream; a packet that cannot be divided/combined is a packet that holds data in the stream form but the data belong to a different data stream, a packet that holds data that are not contiguous, or a packet that holds data in a datagram form that requires to conserve the data boundary. Determining whether a packet can be divided/combined, and dividing/combining packets are described with reference to Figs. 13, and 14.

[0054] With reference Fig. 13, the inner structure of the packet, and dividing/combining operations of the packet are described. Here, for the description purpose, an IP (Internet Protocol) packet generally used in Internet communications as defined by RFC791 is considered. Fig. 13(a) shows the structure of the IP packet, wherein an IP packet 25 includes an IP header 26, and IP data 27. The IP header 26 is an area for holding attribute information of the IP data 27, such as version information of IP protocol used by the IP packet 25, the length of the IP header 26, the length of the IP packet 25 as a whole, a packet ID for identifying the packet, a flag indicating whether the packet can be divided, a division offset value if the packet has been divided for indicating a relative position to an

original packet before being divided, protocol information concerning data carried by the IP data 27, an IP address of the sender of the IP packet 25, and an IP address of the destination of the IP packet 25. The flag of the IP header 26 includes information such as whether the IP packet 25 can be divided, and whether there are subsequent data if the IP packet 25 is an already-divided packet. The protocol of the IP header 26 contains information indicating the kind of protocol applied to the data carried as the IP data 27 such as TCP (Transmission Control Protocol) packet, UDP (User Datagram Protocol) packet, or other protocols indicating the kind of the data stream. The IP data 27 are the data that are to be transmitted.

[0055] A method of dividing and combining an IP packet is defined by RFC791. The outline is described in the following. Whether the IP packet 25 can be divided/combined can be determined based on the version number, the packet ID, the flag, the sender IP address, and the destination IP address contained in the IP header 26. If the flag of the IP header 26 indicates that dividing is possible, the IP packet 25 may be divided. Fig. 13 (b) shows that an IP packet X 25 is divided into two IP packets, namely, an IP packet A 25, and an IP packet B 25.

[0056] With reference to the drawing, an outline of the dividing process performed by the data encoding/dividing means 7 under direction of the data storing/forced transmission directing means 5g is as follows. First, IP data X 27 contained in the IP packet X 25 are divided at an appropriate position into a first part, IP data A 27, and a second part, IP data B 27. Then, based on the IP header X 26 contained in the IP packet X 25, an IP header A 25, and an IP header B 26 are generated. At this time, information contained in the IP header a 26 and the IP header B 26 is almost the same as the contents of

the IP header X 26. Differences are that the flag of the IP header A 26 is set such that there are subsequent data that were divided, and the length of the IP packet as a whole is reduced by the length of the IP data B 27 that were separated; as for the IP header B 26, the flag for showing whether subsequent data are present is set to the same value as the flag of the IP header X 26, the length of the IP packet as a whole is reduced by the length of the IP data A 27 that were separated, and the division offset value is increased by the length of the IP data A 27 that were separated. Finally, the generated IP header A 26 and the IP data A 27 are collected to constitute the IP packet A 25; and similarly, the IP header B 26 and the IP data B 27 are collected to constitute the IP packet B 25.

[0057] Next, a method of combining the IP packets 25 is described with reference to Fig. 13 (c). In the drawing, two IP packets, namely, the IP packet A 25 and the IP packet B 25 are combined to make an IP packet Y 25. Simply put, the IP packets A 25 and B 25 can be combined if they are contiguous packets divided from the singular IP packet 25. In more detail, they can be combined if the version number, the sender IP address, the destination IP address, the protocol, and the packet ID are the same; and the flag of both packets commonly indicates that dividing is possible. If these requirements are met, it is determined that the packets are divided packets from one original packet. Next, the division offset values are compared. A flag with the smaller value (which is assumed to be the IP header A 26, here) indicates that it is the subsequent data after dividing. Further, it is required that the sum of the division offset value and the length of the IP data A be equal to the division offset value of the other IP header B 26. If these conditions are met, it is determined

that the IP packet A 25 and the IP packet B 25 contain contiguous data out of two or more packets generated by dividing the original packet, and that the IP packet A 25 is the first part. Then, the IP packet A 25 and the IP packet B 25 can be combined.

[0058] With reference to the drawing, an outline of the process performed by the data outputting means 11 after receiving data by the data transmitting (sic) means 2 is as follows. Since it has been determined that the IP packet A 25 and the IP packet B 25 can be combined, and the IP packet A 25 is the first part, the IP data A 27 contained in the IP packet A 25 are connected to the IP data B 27 contained in the IP packet B 25, with the IP data A 27 serving as the first part to generate the IP data Y 27. Next, the IP header Y 26 is generated from the IP header A 26 contained in the IP packet A 25, and the IP header B 26 contained in the IP packet B 25. At this time, the information of the IP header Y 26 is the same as the IP header A 26 and the IP header B 26, except for the following differences. That is, the IP packet length of the IP header Y 26 shows a length based on the length of the new IP data Y 27; the flag indicating whether subsequent data are present of the IP header Y 26 has the same value as the flag indicating whether subsequent data are present of the IP header B 26; and the division offset value of the IP header Y 26 has the same value as the division offset value of the IP header A 26. Finally, the generated IP header Y 26 and the IP data Y 27 are collected to constitute the IP packet Y 25.

[0059] Based on the descriptions above, storing the data 20 to the buffer 6 at S32, S65, and S63 in Fig. 12 is described with reference to Fig. 14. In the drawing, the IP packet X 25 is provided by the data inputting means 4 as the data 20, and one or more IP packets 25 are stored in the buffer 6 as the

data 20. Here, the IP packet length of the IP packet 25 as a whole can be determined by examining the IP header 26 of the IP packet 25. Accordingly, when storing two or more IP packets 25 as the data 20 in the buffer 6, it is not necessary to explicitly delimit different packets, but each IP packet can be separated based on the IP packet length.

[0060] Where there are IP packets 25 already stored in the buffer 6, four cases are conceivable as to storing new data 20 to the buffer 6 as shown in Figs. 14 (a) through (d). In the following, the cases wherein there are two IP packets, namely, IP packet A 25, and IP packet B 25, already stored in the buffer 6 as the data 20 are described. This is for description purposes only; the number of IP packets 25 that are already stored in the buffer 6 can be chosen as desired.

[0061] Fig. 14 (a) shows operations carried out in the case wherein the IP packet X 25 contained in the data 20 cannot be combined to either of the IP packets A 25, and B 25; and the size of the IP packet X 25 is smaller than a vacant capacity of the buffer 6. The operations correspond to the processes of S63 (the data 20 are packets that cannot be divided/combined, and all the data 20 can be stored in the buffer 6), and S32 (the data are packets that can be divided/combined, however, there are no packet that can be combined in the buffer 6; and all the data can be stored in the buffer 6) shown in Fig. 12. According to the operations, the entirety of the IP packet X 25 contained in the provided data 20 is stored in the buffer 6 as the data 20 of the buffer 6, following the IP packets A 25 and B 25.

[0062] Fig. 14 (b) shows operations carried out in the case wherein the IP packet X 25 contained in the data 20 cannot be connected to any of the IP packets A 25, and B 25; the size of the IP packet X

25 is greater than the vacant capacity of the buffer 6; and the IP packet X 25 can be divided. The operations correspond to the process of S32 shown in Fig. 12, wherein the data are packets that can be divided/combined, but there is no packet that can be combined in the buffer 6, and a part of the data 20 can be stored. According to the operations, the IP packet X 25 contained in the data 20 is divided into an IP packet X1 25 and another IP packet X2 25, where the size of the IP packet X1 is equal to the remaining capacity of the buffer 6. Then, the IP packet X1 25 is stored in the buffer 6, following the IP packets A 25 and B 25.

[0063] Fig. 14 (c) shows operations carried out in the case wherein the IP packet X 25 contained in the data 20 can be combined to one of the IP packets A 25 and B 25, and all the data 20 as combined can be stored in the buffer 6. The operations correspond to the process S65 shown in Fig. 12, wherein the data are packets that can be divided/combined, there is a packet that can be combined in the buffer 6, and all the data 20 can be stored. Here, it is assumed that the IP packet X 25 can be combined to the IP packet B 25. In this case, according to the operations, the IP packet X 25 contained in the provided data 20 and the IP packet B 25 are combined to generate a new IP packet B+X 25 that is stored in the buffer 6 in place of the IP packet B 25.

[0064] Fig. 14 (d) shows operations carried out in the case wherein the IP packet 25 contained in the data 20 can be combined to one of the IP packets A 25 and B 25, and the combined packet cannot be stored in the buffer due to its size. The operations correspond to the process of S65 shown in Fig. 12, wherein the data are packets that can be divided/combined, a packet that can be combined is

present in the buffer 6, and a part of the data 20 can be stored. Here, it is assumed that the IP packet B 25 and the IP packet X 25 can be combined. In this case, according to the operations, the IP packet X 25 contained in the data 20 is divided into an IP packet X1 25 and an IP packet X2 25, wherein the size of the IP packet X1 25 is equal to the remaining capacity of the buffer 6. Then, the newly generated IP packet X1 25 and the IP packet B 25 already stored in the buffer 6 are combined to generate a new IP packet B+X1 25. The IP packet B+X1 25 is stored in the buffer 6 in place of the IP packet B 25.

[0065] Although the operations above with reference to Figs. 14 (c), and (d) are described in the premise that the IP packet B 25 can be combined to the IP packet X 25, the same results can be obtained if the IP packet to be combined to the IP packet X 25 is the IP packet A 25. Further, it is possible to repeat the present process as far as possible such that a newly combined and generated IP packet 25 is further combined to another IP packet 25 stored in the data 20 in the buffer 6.

[0066] The data storing/forced transmission directing means 5 stores the data 20 in the buffer 6 as the processes of S32, S65, and S63 shown in Fig. 12, and then carries out the following process. That is, if the volume of the data 20 stored in the buffer 6 is equal to or greater than a predetermined value L (Yes at S21); or if the time T has elapsed from the time of storing (Yes at S14), where the volume of the data 20 stored in the buffer 6 is less than the predetermined value L (No at S21), the data 20 stored in the buffer 6 are provided to the data encoding/dividing means 7 (S15), and the buffer 6 is cleared (S16). Further, if the received data 20 cannot be divided (No at S61), and if all the received data 20 cannot be stored in the buffer (No

at S62), the data 20 stored in the buffer 6 are provided to the data encoding/dividing means 7 (S15), and the buffer 6 is cleared (S16) so that the entirety of the received data 20 can be stored in the buffer 6.

[0067] Then, if a part of the received data 20 has not been stored in the buffer 6 (Yes at S33), including the case wherein the received data 20 cannot be divided and cannot be stored in the buffer 6, the unstored part of the data 20 is regarded as newly received data 20 by the data inputting means 4 (S34), and the process is repeated (S31). In the case wherein all the received data 20 are stored in the buffer 6 (No at S33), if new data 20 are present from the data inputting means 4 (Yes at S17), the data 20 are newly received from the data inputting means 4 (S11), and the process is continued. If all the data received are stored in the buffer 6 (No at S33), and there are no new data 20 from the data inputting means 4 (No at S17), the process goes to S14 to determine whether the time T has elapsed since the storing time.

[0068] Following the processes described above directed to the buffer 6 by the data storing/forced transmission directing means 5g, the data encoding/dividing means 7 encodes and divides the data directed by the data storing/forced transmission directing means 5g into N+M packets 21, and transmits the packets from the packet transmitting means 8 to the packet receiving means 9 of the data receiving means 2. The packets 21 received by the packet receiving means 9 are decoded by the data decoding means 10, and provided to the data outputting means 11. Here, the operations of the data encoding/dividing means 7, the packet outputting means, the packet receiving means 9, and the data decoding means 10 are the same as in Embodiment 1;

and descriptions thereof are not repeated. The data outputting means 11 examines the boundary of the packet based on the data 20 provided by the data decoding means 10, and outputs the data 20 separated into packets.

[0069] According to the system of the present embodiment operating as described above, data belonging to different data streams can be collected for encoding, and the size of the buffer 6 can be limited as Embodiment 1; therefore, the processes are efficiently performed. When data belonging to different data streams are provided to the data inputting means 4, data inputting frequency and data volume are increased compared with the case where only one data stream is handled. Accordingly, data transmission opportunities without the time T having been elapsed are increased compared with the case where only one data stream is handled. In this way, a time elapsed before transmission is shortened, and the data communication efficiency is improved.

[0070] Further, the size of an IP packet 25 that consists of two or more IP packets 25 combined is smaller than the total length of the IP packets 25 before being combined by the length of the IP header 16 x n (where, n=the number of the IP packets combined - 1). Accordingly, the operations shown in Figs. 14 (c) and (d), wherein IP packets 25 are combined, can accommodate a greater volume of IP data 27 in the buffer 6 compared with the operations shown in Figs. 14 (a) and (b), where the IP packets are not combined. That is, combining packets, so long as it is possible depending on the contents, when the data storing/forced transmission directing means 5 stores the IP packets 25 received from the inputting means 4 in the buffer 6 as the data 20, increases the amount of IP data 27 that can be provided from the data storing/forced transmission directing means 5 to the

data encoding/dividing means in one shot.

[0071] Although examples of dividing/combining IP packets have been described in above, packet dividing/combining based on an upper protocol is possible. For example, dividing/combining packets by TCP defined by RFC793 is described below.

[0072] Fig. 15 shows the internal structure of a TCP packet, which generally is treated as a kind of an IP packet 25, wherein the IP data 27 contain a TCP header 28 and TCP data 29. Details of the IP header are the same as described with reference to Fig. 13, and descriptions thereof are not repeated. Nevertheless, the protocol of the IP header 26 is fixed to TCP. The TCP header 28 is for storing attribute information of the TCP data 29 such as a sending port number and a destination port number of the TCP packet, a sequence number, an ACK number, and a TCP header length.

[0073] TCP is a protocol that handles a data stream. For this reason, in general, the TCP data 29 held by the TCP packet may be regarded as a part of a data flow that is continuously transmitted from a data transmission end identified by the sender IP address contained in the IP header 26 and the sending port number contained in the TCP header 28 to a data receiving end identified by the destination IP address contained in the IP header 26 and the destination port number contained in the TCP header 28. A position of the TCP data 29 in the data flow can be determined by the size of the TCP data and the sequence number contained in the TCP header 28. Accordingly, if two TCP packets have the same sending IP address and destination IP address in the IP header 26, and the same sending port number and destination port number in the TCP header 28, the two TCP packets are regarded as belonging to the same data stream. Whether the two TCP packets hold

contiguous data in the data stream is determined by the sequence number, and the size of the TCP data. For this reason, packet dividing/combining of the TCP packets can be carried out by performing processes similar to the processes shown in Figs. 13 (b) and (c) for dividing/combining IP packets 25.

[0074] On the protocol properties, the IP packet 25 is generally treated as a "data lump" having a finite size. For this reason, within the "data lump", packet combining and re-dividing are possible as if in the case of a data stream; however, combining and re-dividing are not possible beyond the range. Specifically, this means that combining is possible only between IP packets having the same packet ID. On the other hand, a TCP packet is a part of a "data flow" having an infinite size. Accordingly, if the data sending end (sender IP address and sending port number) and the data receiving end (destination IP address and address port number) are the same, different "data lumps" in the IP level, which is a lower protocol of TCP, that is, data separated by IP packets having different packet IDs can be combined and re-divided. Accordingly, packet dividing and combining in the TCP level are more flexible to restructure the packets than in the IP level, the data communication efficiency can be further improved.

[0075] Embodiment 4

Fig. 16 shows the system configuration according to Embodiment 4. In the drawing, a data transmitting means 1g exchanges data with a data receiving means A 2g via a communication line A 3, and simultaneously exchange data with a data receiving means B 2h via a communication line B 3. In order that data transmitting sources A1 30/A2 30 send data to data receiving sources A1 31a/A2 31b via the data transmitting means 1g and data receiving means A

2g, the data transmitting sources A1 30a/A2 30b provide data 20 to the data inputting means 4 in the form of data streams A1 32a/A2 32b. The data receiving sources A1 31a/A2 31b receive the data 20 from the data outputting means A 11a in the form of the data streams A1 32a/A2 32b. Here, the data stream is a virtual communication line 3 for transmitting data between, e.g., the data transmitting source A1 and the data receiving source A1. Further, in order that data transmitting sources B1 30/B2 30 send data to data receiving sources B1 31c/B2 31d via the data transmitting means 1g and data receiving means B 2h, the data transmitting sources B1 30a/B2 30b provide data 20 to the data inputting means 4 in the form of data streams B1 32c/B2 32d. The data receiving sources B1 31c/B2 31d receive the data 20 from the data outputting means B 11b in the form of the data streams B1 32c/B2 32d. The data transmitting means 1g includes a buffer A 6q for the data receiving means A 2g, and a buffer B 6b for the data receiving means B 2h. Data are stored to these buffers, and the data are provided to the data encoding/dividing means 7b by the data storing/forced transmission directing means 5. Here, other packet transmitting means 8 of the data transmitting means 1g, and other components of the data receiving means A 2a/B 2b, namely, packet receiving means A 9a/B 9b, data decoding means A 10a/B 10b are the same as shown in Fig. 1.

[0076] Next, operations of the data storing/forced transmission directing means 5h according to the present embodiment are described with reference to Fig. 17. The operations start with S70, and data 20 are received (S11). The destination of the data 20 is examined (S71). If the destination of the data 20 is one of the data receiving sources A1 31a, and A2 31b, the data 20 are stored in the buffer A 6a with the information indicating the

destination of the data 20 intact (S72). Next, if the holding time of the oldest data 20 stored in the buffer A 6 exceeds the time T (Yes at S73), the data 20 stored in the buffer A 6a are provided to the data encoding/dividing means 7 as being destined to the data receiving means A 2g, and then the buffer A 6a is emptied (S74). Then, the process returns to S11, wherein the data inputting process is carried out. Otherwise, if the destination of the data 20 is one of the data receiving sources B1 31c, and B2 31d, the data 20 are stored in the buffer B 6b with the information indicating the destination of the data 20 intact (S75). Then, if the holding time of the oldest data 20 stored in the buffer B 6b exceeds the time T (Yes at S76), the data 20 stored in the buffer B 6b are provided to the data encoding/dividing means 7 as being destined to the data receiving means B 2h, and then the buffer B 6b is emptied (S77). Then, the process returns to S11, wherein the data inputting process is carried out.

[0077] The data encoding/dividing means 7 encodes and divide the data provided by the data storing/forced transmission directing means 5h into N+M packets 21, and transmits the packets 21 to the data receiving means A 2g, or the data receiving means B 2h depending on the destination using the packet transmitting means 8. The packets 21 received by the packet receiving means A 9 of the data receiving means A 2g are decoded by the data decoding means A 10, and provided to the data outputting means A 11. Similarly, packets 21 to the data receiving means B 2 are decoded by the data decoding means B 10, and output.

[0078] The data outputting means A 11 examines the contents of the data 20, data 20 addressed to a data receiving source A1 31a are provided to the data receiving source A1 31a, and

data 20 addressed to a data receiving source A2 31b are provided to the data receiving source A2 31b. Similarly, the data outputting means B 11 examines the contents of the data 20, data 20 addressed to a data receiving source B1 31c are provided to the data receiving source B1 31c, and data 20 addressed to a data receiving source B2 31d are provided to the data receiving source B2 31d.

[0079] Operations of examining the data 20 by the data storing/forced transmission directing means 5h, the data outputting means A 11, and the data outputting means B 11 are described with reference to Fig. 15. In the following, data streams A1 32a/A2 32b/B1 32c/B2 32d are virtual communication lines 3 based on TCP; the data 20 that the data storing/forced transmission directing means 5h receives from the data inputting means 4d and the data 20 output by the data outputting means A 11 and B 11 are TCP packets.

[0080] Fig. 15 shows the internal structure of the TCP packet. As described above with reference to the foregoing embodiments, the TCP data 29 held by the TCP packet may be regarded as a part of a data flow that is continuously transmitted from a data transmission end identified by the sender IP address contained in the IP header 26 and the sending port number contained in the TCP header 28 to a data receiving end identified by the destination IP address contained in the IP header 26 and the destination port number contained in the TCP header 28. Here, the destination IP address is generally an identifier of a computer to which the TCP packet is to be transmitted, and the destination port contained in the TCP header 28 identifies a communication target operating in the identified computer. Accordingly, in the present embodiment, without losing the generality, the data receiving means A 2g

and B 2h are differentiated based on the destination IP address; and the data receiving source A1 31a/A2 31b/B1 31c/B2 31d are differentiated based on a set of the destination IP address and IP port address. In this way, the destination of the data 20 is determined based on the IP address and the destination port address contained in the TCP packet.

[0081] According to the operations described above, the data storing/forced transmission directing means 5h determines whether the received data 20 are to be transmitted to the data receiving source A1 31a or the data receiving source A2 31b via the data receiving means A 2g, or to the data receiving source B1 31c or the data receiving source B2 31d via the data receiving means B, by examining the destination IP address contained in the TCP packet received as the data 20 from the data inputting means 4. Further, the data storing/forced transmission directing means 5h directs that the data 20 received from the data inputting means 4 are to be stored in the buffer A 6a if the data are addressed to one of the data receiving sources A1 31a and A2 31b (S72 in Fig. 17), and directs that the data 20 received from the data inputting means 4 are to be stored in the buffer B 6b if the data 20 are addressed to one of the data receiving sources B1 31c and B2 31d (S75 in Fig. 17).

[0082] Similarly, the data outputting means A 11 examines the destination port number contained in the TCP packet received as the data 20 from the data decoding means A 10, determines whether the data 20 are addressed to the data receiving source A1 31a or A2 31b, and outputs the data to the due determination. Similarly, the data outputting means B 11 examines the destination port number contained in the TCP packet of the data 20, determines whether the data 20 are addressed to the data receiving source B1

31c or B2 31, and outputs the data to the due determination.

[0083] Since the operations of the present embodiment are as described above, the data 20 treated on the data stream A1 32a transmitted on the communication line A 3 from the data transmitting source A1 30a to the data receiving source A1 31a, and the data 20 treated on the data stream A2 32b transmitted on the same communication line A 3 from the data transmitting source A2 30b to the data receiving source A2 31b can be stored in the same buffer A 6. Accordingly, data belonging to different data streams can be collectively encoded by the data encoding/dividing means 7, which improves the efficiency of encoding and transmission.

[0084] Fig. 18 shows a specific image of the data transmitting means 1, the data receiving means 2, and the communication line 3 described above. In the drawing, two communication apparatuses A 40a, and B 40b are connected to the Internet 42 using a communication interface B connected to the packet transmitting means 8 and the packet receiving means 9; and are connected to a LAN 43 via a communication interface A connected to the data inputting means 4 and the data outputting means 11, such that a data communication relaying process is carried out between the two LANs 43 using the Internet 42. One of the communication apparatuses A and B functions as the data transmitting means 1, and the other functions as the data receiving means 2. The Internet 42 serves as the communication line 3.

[0085] The communication apparatuses A 40a and B 40b improve the quality and efficiency of the data communications of the communication line B 3 connected to the communication interface B as described above. Communications between the two LANs 43 connected by the Internet 42 tend to have a great

communication delay, distribution of communication times, and data lost; by applying to this, a high quality and high efficiency transmission is possible.

[0086] Data dividing/combining of the same data stream as described in Embodiment 3 can be applied to applications. Fig. 19 shows the system structure, wherein a computer system 44 includes an application 45, a first data communication means 46, and a second data communication means 47 that further includes the data transmitting means 1 and the data receiving means 2 connected to the communication line 3.

[0087] Operations of various components are as follows. The application 45 exchanges data with the first data communication means 46 for data communications. The first data communication means 46 operates the second data communication means 47 as an alternate of the communication line 3, and exchanges packets with the second data communication means. The second data communication means 47 regards the exchange of the packets with the first data communication means 46 as exchange of data, and exchanges the packet with the communication line by operations of the data transmitting means 1 and the data receiving means 2. The internal operations of the data transmitting means 1 and the data receiving means 2 have been described, and the descriptions are not repeated. In this way, the quality and efficiency of the data communication of the communication line 3 are improved by the second data communication means 47; by using, for example, a general communication means such as TCP/IP protocol stack as the first data communication means 46, data communications can be carried out as if connection is made using a higher quality communication line 3 without modifying the first data communication means.

[0088]

[Effect of the Invention] As described above, according to the present invention, the time during which the data are held in the buffer is limited, and the length of dividing/encoding the data is made variable by the communication line information, the transmission can be completed in a short time, and since the transmission is carried out at a predetermined length, overhead increase is avoided; therefore, the high-quality and highly efficient data transmission is made possible.

[Brief Description of Drawings]

Fig. 1 shows the system configuration according to Embodiment 1 of the present invention.

Fig. 2 shows operations of the system according to Embodiment 1.

Fig. 3 is a flowchart showing operations of mainly the data storing/forced transmission directing means according to Embodiment 1.

Fig. 4 shows operations of the data encoding/dividing means according to Embodiment 1.

Fig. 5 shows operations of the data decoding/dividing means according to Embodiment 1.

Fig. 6 is a flowchart showing operations of mainly another data storing/forced transmission directing means.

Fig. 7 is a flowchart showing operations of mainly another data storing/forced transmission directing means.

Fig. 8 is a flowchart showing operations of mainly the data storing/forced transmission directing means according to Embodiment 2 of the present invention.

Fig. 9 is a flowchart showing operations of mainly another data storing/forced transmission directing means according to Embodiment 2.

Fig. 10 shows another system configuration

according to Embodiment 2.

Fig. 11 shows another system configuration according to Embodiment 2.

Fig. 12 is a flowchart showing operations of mainly the data storing/forced transmission directing means according to Embodiment 3 of the present invention.

Fig. 13 shows the internal structure of the packet, and dividing/combining packets according to Embodiment 3.

Fig. 14 shows different cases of storing the data in the buffer.

Fig. 15 shows the internal structure of a TCP packet.

Fig. 16 shows the system configuration according to Embodiment 4 of the present invention.

Fig. 17 is a flowchart showing operations of mainly the data storing/forced transmission directing means according to Embodiment 4.

Fig. 18 shows an example of a specific connection formation.

Fig. 19 shows an example of applying the present invention to an application.

Fig. 20 shows an example of the configuration of a first conventional data transmission system.

Fig. 20 shows an example of the configuration of a second conventional data transmission system.

[Description of Notations]

1, 1b, 1c, 1d, 1g: data transmission means

2, 2a, 2b, 2c, 2e, 2f, 2g, 2h: data receiving means(A), (B)

5, 5b, 5c, 5d, 5e, 5f, 5g, 5h: data storing/forced transmission directing means

6, 6a, 6b: buffer (A), (B)

7, 7b: data encoding/dividing means (A), (B)

10 data decoding means (A), (B)

S14: step of determining time elapsed

[end]